Interpreting prepositions physically

J.K. Kalita
Department of Computer Science
University of Colorado
Colorado Springs, CO 80933

N. I. Badler
Department of Computer & Information Science
University of Pennsylvania
Philadelphia, PA 19104

Abstract

We develop representations for locative and path specifying prepositions emphasizing the implementability of the underlying semantic primitives. Our primitives pertain to mechanical characteristics such as geometric relationships among objects, kinematic or motional characteristics implied by prepositions. The representation along with representation for action verbs along similar lines, have been used to successfully animate the performance of tasks underlying natural language imperatives by "human" agents.

1 Introduction

Consider the imperatives

- Carefully place the block inside the box.
- Gently put the block on the table.
- Place the ruler across the table.
- Slowly roll the ball across the rug.

Here, we have three action verbs in the sentences above: *put, place* and *roll.* There are three prepositions *inside, on* and *across. Across* is used in two different senses—static as well as and dynamic. *Inside* and *on* are used in static senses in the first two sentences. *Inside* in the first sentence expresses a relation in term of the box's location and geometry; *across* gives a relation in terms of the reference object's location and geometry (co-linearity with the longitudinal axis).

2 Representational Features

We have discussed the representation of the meanings of action verbs in [Kalita 90a, Kalita90b, Badler 90] in terms of physical features—geometric relationships, aspectual considerations such as repetitiveness of sub-actions and definedness of termination conditions, kinematic or motional attributes, dynamic or force-related features.

Geometric constraints provide information regarding how one or more objects or sub-parts of objects relate to one another in terms of physical contact, absolute or relative location, inter-object distance, absolute or relative orientation or path of motion.

Constraints are of two types. *Positional constraints* refer to a situation in which a $0\cdots3$- dimensional object is constrained to a $0\cdots3$- dimensional region of space. For example, in order to execute the command *Put the ball on the table*, an arbitrary point on the surface of the ball has to be brought in contact with (or constrained to) an arbitrary point on the surface of the table. In the action underlying the imperative *Put the block in the box* where one needs to constrain the block (or the volume occupied by the block) to the interior volume of the box. *Orientational constraints* are useful in situations such as representing the meaning of the preposition *across* in the sentence *Place the ruler across the table.* The interpretation of the preposition involves several components, one of which requires that the longitudinal axis of the ruler and the longitudinal axis of the table top be perpendicular to each other.

Verbs dealing with constraints can be classified considering whether they denote establishment, removal or maintenance of (existing) geometric constraints. There are verbs whose central actions require that constraints established continue to hold: *attach, hold, fix, grasp.* The central actions of other verbs require that already existing constraints *cease* to hold. Examples include: *detach, disconnect, disengage, release.*

3 Prepositional Phrases

Locative modifiers specify where in the workspace the action is performed. A location can be specified with respect to one object or its features (e.g., *in, inside, behind,* and *against*). A location can also be specified with respect to two or more objects, their sub-parts or features (e.g., *between* and *among*).

Specification of path is a part of kinematic specification of a motion or an action. A complete definition of path requires specification of source, destination and path geometry. Prepositions in this category include *from, to, along, around, round* and *across.*

4 Locative Prepositions

Although, several of detailed studies of English prepositions are available (e.g., [Funk 53]), very few provide meanings in a precise, implementable format (except

[Badler 75] and [Gangel 84]). We use an extended version of Talmy's [1983] schema approach made suitable for computational purposes. We have also been influenced by [Herskovits 86].

We organize the meaning of a preposition such as *on* in the following way:

on

on_1 on_2 $\cdots$ on_n

At the top level, we have the preposition itself. The second level contains a list of senses of the preposition for which we have lexical entries.

Generally speaking, one might be able to define a general meaning which resides in the root and then provide inference procedures or heuristics for deriving representation for the senses. However, these inference procedures or heuristics are still neither well understood nor exhaustively enumerated although Herskovits discusses them in a general manner. For the purposes of obtaining lexical entries in this paper, we consider one sense of each preposition to keep our discussions simple.

The structure used for representing meanings of locative prepositions is in terms of a representation template: *geometric-relation*. A geometric relation is specified in two ways. The first, which represents a simple geometric relation is a frame with the slots:

geometric-relation:
    spatial-type:
    source-constraint-space:
    destination-constraint-space:
    distance:
    weight:
    selectional-restrictions:

The second is for more complex cases which are composed of two or more geometric relations to be satisfied simultaneously:

geometric-relation:
    { g-union g-relation-1 g-relation-2 $\cdots$ g-relation-n }

where *g-relation-i* is simple or complex.

4.1 A Representation for *on*

The sense of *on* we are interested in is seen in the sentence *Put the block* on *the table*. The relevant use type of *on* from Herskovits is *spatial entity supported by physical object*. A handbook describes it as *in contact with upper surface of; above and supported by* [Funk 53].

The relations implied in this meaning of *on* are: *contact, support* and *above*. The relationship of support is difficult to define. Fahlman devised complex heuristic and mathematical tests to determine if an assembly of blocks can be supported by another [1974]. We do not define *support* in such complex manners. Our representation of *support* is indirect and symbolic. We describe this meaning or sense of *on* as

on (X,Y) ⟵
    geometric-relation:
        spatial-type: positional
        source-constraint-space: any-of
          (self-supporting-spaces-of (X))
        destination-constraint-space: any-of
          (supporter-surfaces-of (Y)))
    selectional-restrictions:
        horizontal-p (destination-constraint-space)
        equal ((direction-of (normal-to
          destination-constraint-space), "global-up")
        free-p (destination-constraint-space)

It is impossible to capture all selectional restrictions or applicability conditions associated with any word in a natural language [Winograd 80]. No matter what conditions are imposed, one can always fabricate an odd context in which the condition does not hold.

Given a geometric object, the geometric function *self-supporting-spaces-of* obtains a list containing surfaces, lines or points on the object on which it can support itself. For example, a cube can be supported on any of its six faces, and a sphere on any point on its surface. The function *supporting-surfaces-of* takes an object as an argument and finds outer surfaces on it on which other objects can be supported. To do so, it may find out if there are parts whose function is primarily to support other objects (e.g., the top surface of a table), failing which it finds out if there are parts which are normally horizontal, or are horizontal at the current time.

The definition for *on (X, Y)* given above specifies all three relations: *contact, support* and *above*. *Contact* is specified in terms of a geometric relation between a source and a destination constraint space. *Support* is specified indirectly by requiring that the *source-constraint-space* is a *self-supporting-space* of *X* and that the *destination-constraint-space* is a *supporter-surface-of* of *Y*. *Above* is specified in terms of restrictions we impose on the direction of the normal to the *destination-constraint-space* and by requiring that the *destination-constraint-space* is horizontal.

4.2 A Representation for *in*

The handbook defines this meaning as *within the bounds of, contained in or included within* [Funk 53]. According to Herskovits this use type for *in* is *spatial entity in a container*. We specify this meaning of *in* as

in (X,Y) ⟵

geometric-relation:
    spatial-type: positional
    source-constraint-space: volume-of (X)
    destination-constraint-space: interior-of (Y)
    selectional-restrictions:
        or (container-p (Y),
            container-p (any-of (sub-parts-of (Y))))
        size-of (X) $\leq$ size-of (Y)
        normally-oriented (Y)

A *container* is an object which can hold one or more objects such that the object is "surrounded by" the volume defined by the boundaries of the container. It is a concept which is difficult to define clearly, although heuristics can be devised to recognize whether an object is a container. For our purposes, if an object or any of its part(s) can work as container(s), we will label it (them) as such in the *function* slot in its representation. The second condition is due to Cooper [Cooper 68]. The third condition is due to Herskovits who explains its necessity by stating that the sentence *The bread is in the bowl* is pragmatically unacceptable if the bowl is upside down and covers the bread under it [Herskovits 86].

5 Projective Prepositions

The prepositions considered in this section are called projective because they define directions about an object, and specify the location of another object in relation to these directions [Herskovits 86].

First, we define a global coordinate system, assumed to be placed in front of the work area.

+ve Z	Aligned against gravity; also called the *global-up* direction.
-ve Z	Same as the direction of gravity; also called *global-down* axis.
+ve X	Also called *global-right*.
-ve X	It is the *global-left* direction.
+ve Y	It is the *global-front* direction.
-ve Y	This is the *global-back* direction.

These six axes are called *half axes* [Herskovits 86]. A full axis is a two-directional line constituted from a pair of companion half axes. We define a *half axial plane* as a plane which passes through one half axis and a full axis. Two companion half axial planes constitute a *full axial plane*. For instance, *left-right* full axial plane is a true planar surface extending to infinity in all directions and passing through the left-right and up-down full axes.

We also establish a reference system rooted at the center of volume of the object under consideration. Assuming we have one animate actor and the objects manipulated are inanimate, it is sufficient to consider the *canonical encounter situation* for this reference systems.
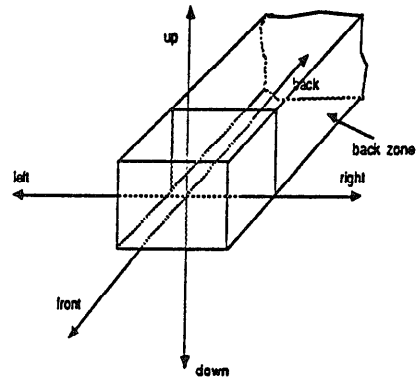

Figure 1: The Back Zone

Here, it is assumed that the observer is situated at some point directly in front of the object.

Douglas et al characterize 2-D specialized regions defined by the area occupied by an object for defining projective prepositions [Douglas 87]. They define pairs such as *front* and *back* in terms of two-dimensional regions obtained by projecting the area occupied by the objects. We treat these projective prepositions in terms of three dimensional volumes. For example, to define *behind* an object, we construct a volume called the back zone of the object by doing the following:

1. Draw two tangent planes to the object parallel to the front-back full axial plane.
2. Draw two tangent planes to the landmark object parallel to the left-right full axial plane.
3. The contiguous volume bounded by these four planes and the volume of the object forms the back zone of the object. This is also shown in figure 1.

5.1 *Behind*

Having understood the concept of *back zone*, we define the concept of *behind*. The definition requires that we make the back-front axial planes of Y and X coincident, and that the centroid of the first object be contained within the *back-zone* of Y. The lexical entry is

behind (X,Y) ⟵
    { g-union
    { geometric-relation:
        spatial-type: positional
        source-constraint-space: back-half-axial-plane (X)
        destination-constraint-space:
            back-half-axial-plane (Y))) }
    { geometric-relation:
        spatial-type: positional
        source-constraint-space: centroid (X)
        destination-constraint-space:back-zone (Y))) } }

In other words, object Y is properly behind object X if 1) behind half axial planes of X and Y coincide, 2) the centroid of X is in the back-zone of Y. Problems due to relative sizes of the objects is taken care of by the underlying constraint satisfaction techniques and the use of *weight* slot in the template for constraint representation. Meanings of a few other projective prepositions such as *in front of* and *left of* can be defined in a similar manner.

6   Three-object prepositions

6.1   *Between*

In order to define *between*, we define a volume called *between-zone (X, A, B)*. X is the object to be placed between A and B. We draw a plane P passing between the centroids of A and B, and the "global-up" axis. The two points and the line define the unique plane P. Next, we define the *projection-zones* of the two objects:

- Projection-zone(A) is in the direction from A to B.
- Projection-zone(B) is in the direction from B to A.

The intersection of these two projection zones obtains the *between-zone (X, A, B)*.

between (X,A,B) ⟵
      geometric-relation:
            spatial-type: positional
            source-constraint-space: volume-of (X)
            destination-constraint-space:between-zone (A,B)

7   Path prepositions

Specification of path needs, at least, the following components: source, destination, and path geometry. All components can be seen in the following example (slightly contrived): *Move the block from one end of the table to the other along an edge.*

We represent a path in terms of a structure called *path-specification* with the slots:

path-specification:
      source:
      destination:
      path-geometry:

This structure is used for describing meanings of prepositions which specify paths which may be partial.

7.1   *Across*

Across has two types of meanings—dynamic and static (locative) meaning. The dynamic meaning implies a journey across an object, whereas the static meaning implies a location between two lines (edges) perpendicular to them and touching, and (possibly) extending beyond them. The dynamic sense of *across* is seen in:

- Roll/Slide/Move the block/ball across the board.

The dynamic sense of *across* specifies all three components required for path specification.

across (X, Y) ⟵
      path-description
      source: any-of (exterior-edges-of (Y, parallel-to
                            (longitudinal-axis (Y))))
      destination:any-of (exterior-edges-of (Y, parallel-to
                            (longitudinal-axis (Y))))
      path-geometry: straight-line
      selectional-restrictions:
            destination $\neq$ source
            has-axis (X, longitudinal)
            angle-between
                  (path-geometry, longitudinal-axis (Y), 90°)
            length (Y) $\geq$ width (Y)
            length (Y) $>$
                  (dimension-of X (along-direction
                        (longitudinal-axis Y))

The *longitudinal axis* of an object is the axis along which the *length* of an object is measured. There are a number of selectional restrictions imposed on the objects X and Y also. For example, the reason for the fourth selectional restriction can be gauged from the two phrases: *across the road* and *along the road*.

8   Processing a sentence

Let us take the sentence *Put the block on the table.* The lexical entry for *put* specifies the achievement of a geometric relationship between an object and a location specified by a PP. The meaning of the verb is specified in terms of a yet-unspecified geometric relation between two objects. The preposition *on* along with the objects involved leads to the sense that deals with support.

A bottom-up parser returns the logical meaning representation as (*put you block-1 (on block-1 table-1)*). In this representation, the verb *put* takes three arguments: a subject, an object and the representation for a locative expression. *block-1* and *table-1* are objects in the world determined to be the referents of the noun phrases. The logical representation has *you* as the value of the subject since the sentence is imperative.
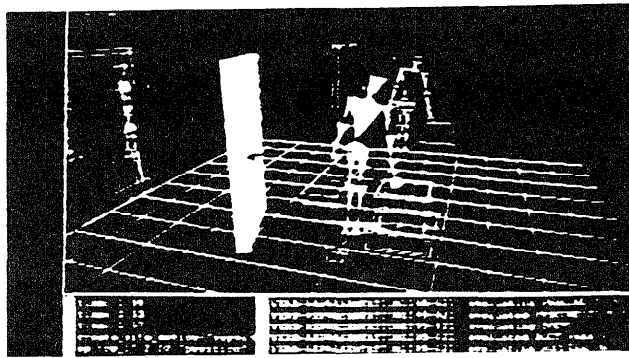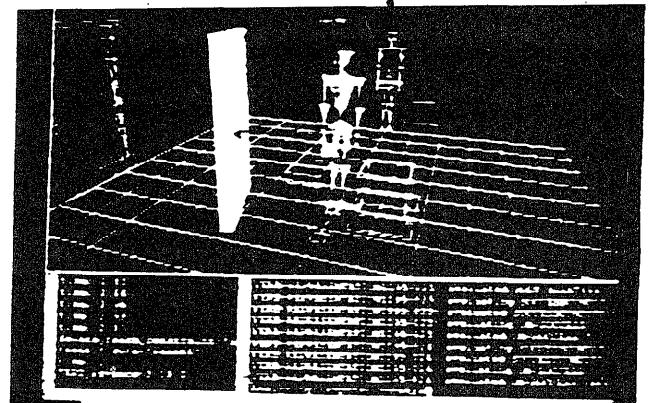
1: Agent bends over to grasp the block
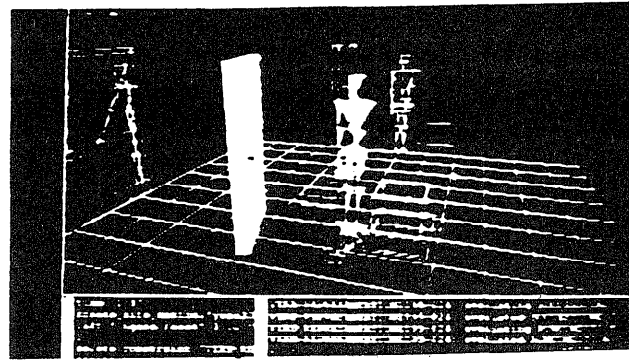
3: The block is on the table
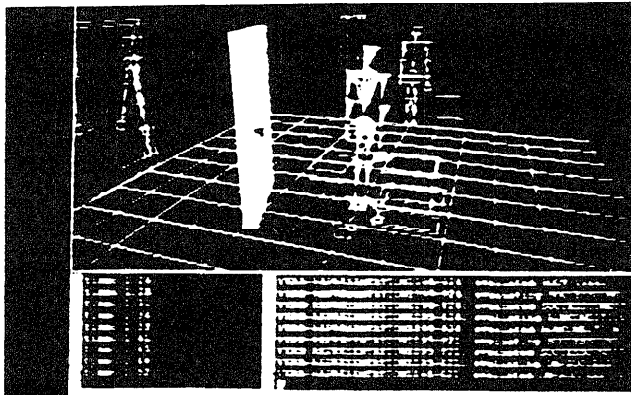
2: The block is in the agent's hand which is in the mid-air

4: The agent is back in normal upright position

Figure 2: Animating *Put the block on the table*

Now, to obtain the intermediate meaning representation, the arguments of *put* in the logical representation are matched with the arguments in the lexical entry for *put* given below. The representation for a verb has a slot called *kernel-actions* which stores its essential meaning. This slot can be filled by a specification of a geometric constraint, or kinematic or motional specification (for motion-primary verbs such as *move, roll*) and dynamic specification (for force-primary verbs such as *push, pull, hit*). Our representation for *put* is

put (l-agent, l-object, l-locative) ⟵
    agent: l-agent
    object: l-object
    kernel-actions:
        geometric-constraint:
            execution-type:achieve
            geometric-relation: l-locative

This lexical entry has three arguments. After matching, *l-agent* has the value *you*, *l-object* has the value *block-1*, and *l-locative* has the value (*on block-1 table-1*). The value of the *geometric-relation* slot (of the *kernel-actions* slot in the representation) is filled in by the semantic representation for the *l-locative* argument which is created recursively. Thus, the intermediate meaning representation is

agent: you
object:block-1
kernel-action:
    geometric-constraint:
        execution-type: achieve
        geometric-relation:
            spatial-type: positional
            source-constraint-space: any-of
                (self-supporting-spaces-of (block-1))
            destination-constraint-space: any-of
                (supporter-spaces-of (table-1)))
    selectional-restrictions:
        horizontal-p (destination-constraint-space)
        equal (direction-of (normal-to
            destination-constraint-space), "global-up")
        free-p (destination-constraint-space)

In order to execute the action dictated by this sentence, the program looks at the knowledge stored about the block to find a part of the block on which it can support itself. It observes that it can be supported on any one of its faces and no face is more salient than any other. A cube (the shape of the block) has six faces and one is chosen randomly as the support area. Next, the program consults the knowledge stored about the table and searches for a part or feature of the desk which can

be used to support other objects. It gathers that its function is to support "small" objects on topside. This top surface is also horizontal. As a result, finally, the system concludes that one of the sides of the cube has to be brought in contact with the top of the table.

The final meaning for the sentence obtained is

```
agent:  you
object: block-1
kernel-actions:
    geometric-constraint:
        execution-type:     achieve
        geometric-relation:
            spatial-type:                   positional
            source-constraint-space: block-1•side-2
            destination-constraint-space: table-1•top
```

block-1•side-2 represents a specific face of a specific block. *table-1•top* represents the top surface of a specific table. This final representation is then sent to a planner [Jung 91] which produces a plan for performing the task by an animated agent in a given workspace. The plan is taken up by a simulator [Esakov 90] which establishes connection with a graphical animation package [Phillips 88] and produces an animation of the task performance. We show a few snapshots below. The block is initially sitting on top of a closed box. The agent reaches for it with his right hand, grasps it, moves it to near the top of a table to his left, places it on the table, and moves his hand back.

9   Conclusions

In this paper, we discussed the representation of meanings of some locative and path specifying prepositions. We emphasize the importance of geometric information such as axes of objects, location of objects, distance or angle between objects, path of object motion, physical contact between objects, etc., in the meaning representation of prepositions. Elsewhere, we demonstrate that such geometric considerations are important for not only representation for prepositions, but also verbs and adverbs [Kalita 90a]. Our representations are geared toward obtaining an implementable semantics of natural language words used to describe actions. We demonstrate the sufficiency and usefulness of our representation by establishing connection with a graphical animation package and by driving task performance.

References

[Badler 75] Badler, N. *Temporal Scene Analysis: Conceptual Description of Object Movements.* Univ. of Penn., Dept. of Computer and Information Science, TR 76-4.

[Badler 90] Badler, N., B. Webber, J. Kalita, and J. Esakov. Animation from instructions. In Badler, N., D. Zeltzer, and B. Barsky (editors), *Making Them Move: Mechanics, Control, and Animation of Articulated Figures.* Springer Verlag, New York.

[Cooper 68] Cooper, G. *A semantic Analysis of English Locative Prepositions*, BBN Report No 1587, Springfield, VA: Clearinghouse for Federal Scientific and Technical Information.

[Douglas 87] Douglas, S. and D. Novick. Consistency and Variance in Spatial Reference. In *Proceedings of the Ninth Annual Cognitive Science Society Meeting*, pages 417-426.

[Esakov 90] Esakov, J. and N. Badler. An Architecture for High-Level Human Task Animation Control. In Fishwick, P. and Modjeski, R. (editors), *Knowledge-Based Simulation: Methodology and Applications.* Springer Verlag, New York.

[Fahlman 74] Fahlman, S. A Planning System for Robot Construction Tasks. *Artificial Intelligence* 5:1-49.

[Funk 53] Funk and Wagnallis Editorial Staff. *Standard Handbook of Prepositions, Conjunctions, Relative Pronouns and Adverbs.* Funk and Wagnallis Co., New York.

[Gangel 84] Gangel, J. A Motion Verb Interface to a Task Animation System. Master's thesis, Dept. of Computer and Information Science, Univ. of Pennsylvania.

[Herskovits 86] Herskovits, A. *Language and Spatial Cognition. Studies in Natural Language Processing.* Cambridge University Press, Cambridge, England.

[Jung 91] Jung, M. Posture Planning for Animation of Human Tasks in a Physical World, Ph.D. thesis proposal, Univ. of Pennsylvania.

[Kalita 90a] 24 Kalita, J. *Natural Language Control of Task Performance in a Physical Domain*, Ph.D. thesis, Univ. of Pennsylvania.

[Kalita90b] Kalita, J., and N. Badler. A Semantic Analysis of a Class of Action Verbs Based on Physical Primitives, *Annual Meeting of the Cognitive Science Society*, pp. 412-419.

[Phillips 88] Phillips, C. and N. Badler. JACK: A Toolkit for Manipulating Articulated Figures. In *Proceedings of ACM SIGGRAPH Symposium on User Interface Software*, pages 221-229. Banff, Alberta.

[Talmy 83] Talmy, L. How Language Structures Space. In Pick, H. and L. Acredols (editors), *Spatial Orientation: Theory, Research and Application*, pages 225-282. Plenum Press, New York.

[Winograd 80] Winograd, T. What Does it Mean to Understand Language? *Cognitive Science* 4:209-241.